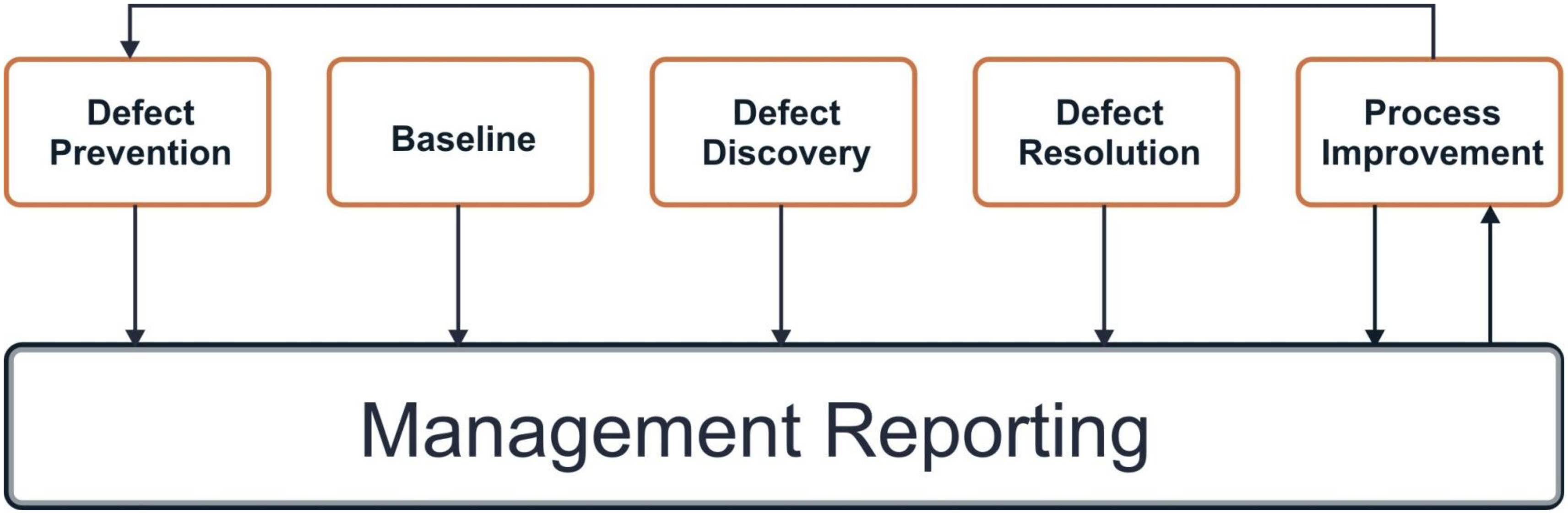
Review or research in software defect reporting

Anna Gromova, Exactpro

Open Access Quality Assurance & Related Software Development for Financial Markets
Tel: +7 495 640 2460, +1 415 830 38 49
www.exactpro.com

Defect management



Defect Management

Areas of research in defect management [1]:

- automatic defect fixing
- automatic defect detection
- triaging defect reports
- quality of defect reports
- metrics and predictions of defect reports

1] Johnatan D. Strate, Phillip A. Laplante "A literature review of research in software defect "

Open Access Quality Assurance & Related Software Development for Financial Markets
www.exactpro.com
Tel: +7 495 640 24 60 , +1 415 830 38 49

Automatic defect fixing

Tasks:

- automatic fixing of unit-tests
- automatic fixing of found detects

Open Access Quality Assurance & Related Software Development for Financial Markets
www.exactpro.com
Tel: +7 495 640 24 60 , +1 415 830 38 49

Automatic defect fixing

Genetic programming

- Evolve both programs and test cases at the same time [1]
- Avoid defects and retain functionality [2]

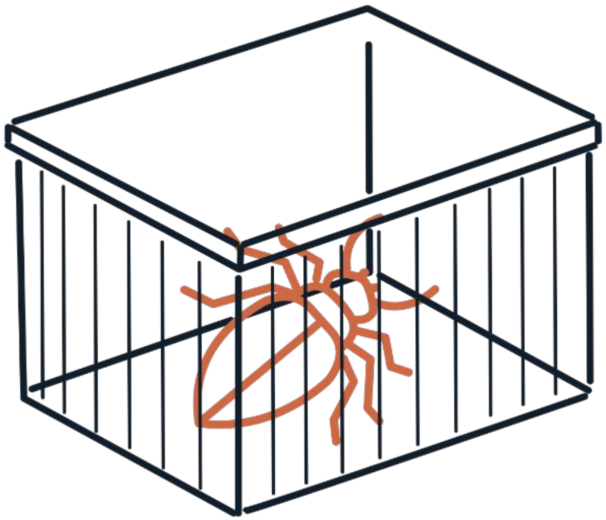
[1] A.Arcuri, X. Yao "A novel co-evolutionary approach to automatic software bug fixing"
[2] W. Weimer, T. Nguyen, C. Le Goues, and S. Forrest, "Automatically finding patches using genetic programming"

Automatic defect fixing

exactpro
EXITUS ACTA PROBAT

SBSE

- Searching code for possible defects [1]
- Adaptive bug isolation [2]



[1] M. Harman, P. McMinn, J. de Souza, and S. Yoo, “Search based software engineering: Techniques, taxonomy, tutorial”,
”M. Harman, “Software engineering meets evolutionary computation”
[2] P. Arumuga Nainar and B. Liblit, “Adaptive bug isolation”

Automatic defect fixing

Tools:

- Co-evolutionary Automated Software Correction [1]
- AutoFix-E / AutoFixE2 [2]
- ReAssert [3]
- GenProg [4]

[1] J. L. Wilkerson and D. Tauritz, “Coevolutionary automated software correction”
[2] Y. Wei, Y. Pei, C. A. Furia, L. S. Silva, S. Buchholz, B. Meyer, and A. Zeller, “Automated fixing of programs with contracts”,
Y. Pei, Y. Wei, C. Furia, M. Nordio, and B. Meyer, “Code-based automated program fixing”
[3]B. Daniel, V. Jagannath, D. Dig, and D. Marinov, “Reassert: Suggesting repairs for broken unit tests”
B. Daniel, T. Gvero, and D. Marinov, “On test repair using symbolic execution”
[4] . Le Goues, T. Nguyen, S. Forrest, and W. Weimer, “Genprog: A generic method for automatic software repair”

Automatic defect defection

Tasks:

- Search defects [1]
- Predict defects [2]
- Predict number of defects [3]
- Predict post-release defects[4]

[1] C. C. Williams and J. K. Hollingsworth, “Automatic mining of source code repositories to improve bug finding techniques”; J. DeMott, R. Enbody, and W. Punch, “Towards an automatic exploit pipeline”

[2] R. Moser, W. Pedrycz, and G. Succi, “A comparative analysis of the efficiency of change metrics and static code attributes for defect prediction”; S. Kim, T. Zimmermann, E. J. Whitehead, Jr., and A. Zeller, “Predicting faults from cached history”; A. E. Hassan, “Predicting faults using the complexity of code changes”

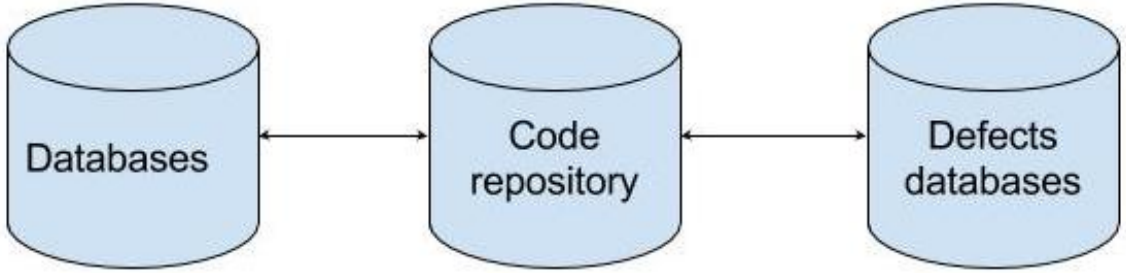
[3] C.-P. Chang, J.-L. Lv, and C.-P. Chu, “A defect estimation approach
for sequential inspection using a modified capture-recapture model”, R. Bucholz and P. Laplante, “A dynamic capture-recapture model for software defect prediction”

[4] T. Zimmermann, R. Premraj, and A. Zeller, “Predicting defects for eclipse”, N. Nagappan, T. Ball, and A. Zeller, “Mining metrics to predict component failures”; N. Fenton, M. Neil, W. Marsh, P. Hearty, D. Marquez, P. Krause, and R. Mishra, “Predicting software defects in varying development lifecycles using bayesian nets”

Automatic defect defection

Tools:

- Linkster [1]
- BugScout [2]

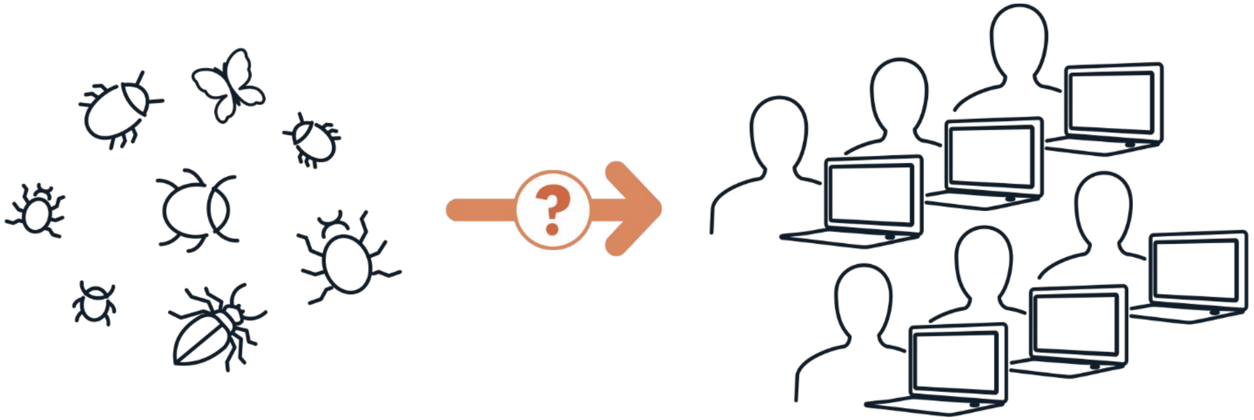


[1] A. Bachmann, C. Bird, F. Rahman, P. Devanbu, and A. Bernstein, “The missing links: Bugs and bug-fix commits”
[2] A. T. Nguyen, T. T. Nguyen, J. Al-Kofahi, H. V. Nguyen, and T. Nguyen, “A topic-based approach for narrowing the search space of buggy files from a bug report”

Triaging defect reports

Tasks:

- Classify defect reports
- Detecting duplicates
- Automatic assignment



Open Access Quality Assurance & Related Software Development for Financial Markets
www.exactpro.com
Tel: +7 495 640 24 60 , +1 415 830 38 49

Triaging defect reports

- Classify defect reports:

- Defect or non-defect [1]
- Security risk [2]
- Crash-types [3]

[1] G. Antoniol, K. Ayari, M. Di Penta, F. Khomh, and Y.-G. Guéhéneuc, “Is it a bug or an enhancement?: A text-based approach to classify change requests”
[2] M. Gegick, P. Rotella, and T. Xie, “Identifying security bug reports via text mining: An industrial case study”
[3] F. Khomh, B. Chan, Y. Zou, and A. Hassan, “An entropy evaluation approach for triaging field crashes: A case study of mozilla firefox”

Triaging defect reports

Reasons for duplicates [1]:

- unexperienced users,
- poor search features,
- multiple failures - one defect,
- accidental resubmission

[1] N. Bettenburg, R. Premraj, T. Zimmermann, and S. Kim, “Duplicate bug reports considered harmful really?”

Triaging defect reports

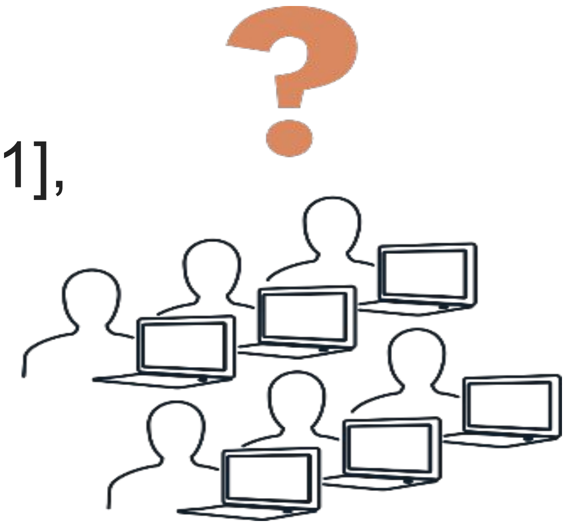
Detecting duplicates:

- NLP + information extraction [1]
- Textual semantic + clustering [2]
- N-gram-based model [3]
- Keywords repository [4]

[1] X. Wang, L. Zhang, T. Xie, J.Anvik,and J.Sun, “An approach to detecting duplicate bug reports using natural language and execution information”
[2] N. Jalbert and W. Weimer, “Automated duplicate detection for bug tracking systems”
[3] A. Sureka and P. Jalote, “Detecting duplicate bug report using character n-gram-based features”
[4] S. Tan, S. Hu, and L. Chen, “A framework of bug reporting system based on keywords extraction and auction algorithm”

Automatic assignment:

- Predict developer : text categorization [1], SVM [2], information retrieval [3]
- Recommenders: machine learning [4]



[1] D.Čubranić,“Automatic bug triage using text categorization”
[2] Z. Lin, F. Shu, Y. Yang, C. Hu, and Q. Wang, “An empirical study on bug assignment automation using chinese bug data,”
[3] D. Matter, A. Kuhn, and O. Nierstrasz, “Assigning bug reports using a vocabulary-based expertise model of developers”
[4] J. Anvik, L. Hiew, and G. C. Murphy, “Who should fix this bug?”

Automatic defect fixing

exactpro EXITUS ACTA PROBAT

Tools:

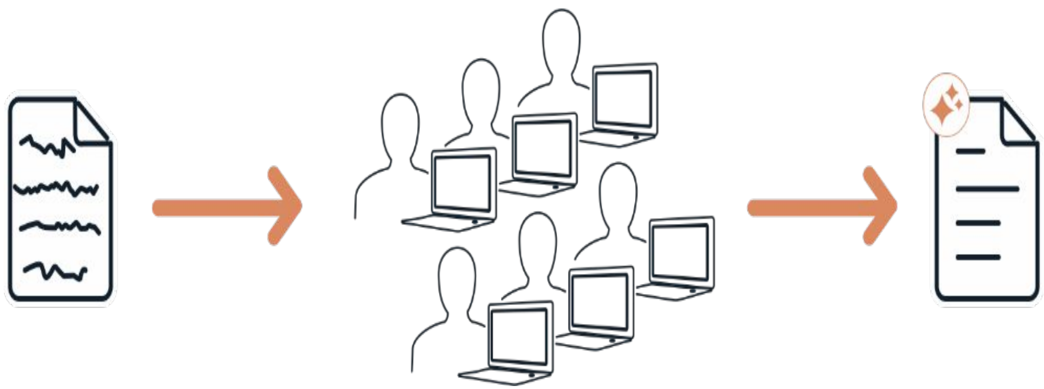
- Bugzie [1]
- DREX [2]

[1] A.Tamrawi,T.T.Nguyen,J.M.Al-Kofahi,and T.N.Nguyen,"Fuzzy set and cache-based approach for bug triaging,"
[2] W. Wu, W. Zhang, Y. Yang, and Q. Wang, "Drex: Developer recommendation with k-nearest-neighbor search and expertise ranking"

Quality of defect-reports

Tasks:

- Surveying Developers and Testers
- Improving defect reports



Open Access Quality Assurance & Related Software Development for Financial Markets
www.exactpro.com
Tel: +7 495 640 24 60 , +1 415 830 38 49

Quality of defect-reports

Results of survey [1]:

SELECTION% AND USEFULNESS OF ITEMS

Item	Selected	Slightly Useful	Fairly Useful	Quite Useful	Very Useful
Bug title	64%	23%	34%	26%	17%
Component / module	77%	12%	16%	40%	32%
Configuration	82%	7%	31%	36%	26%
Error reports	70%	13%	17%	38%	31%
Expected behavior	69%	2%	18%	35%	45%
Hardware context	34%	40%	44%	8%	8%
Observed behavior	77%	5%	7%	28%	60%
Operating data	89%	6%	20%	26%	48%
Part of the application	92%	3%	6%	25%	66%
Product information	64%	13%	30%	26%	32%
Contact information	58%	33%	30%	19%	19%
Screenshots	95%	4%	19%	27%	50%
Severity of the bug	54%	30%	45%	22%	2%
Software context	57%	31%	40%	21%	7%
Stack trace	70%	8%	17%	35%	40%
Steps to reproduce	97%	0%	0%	3%	97%
Test cases, test scripts	47%	9%	40%	26%	26%
User input	69%	4%	22%	51%	24%

[1] E. I. Laukkanen and M. V. Mantyla, “Survey reproduction of defect reporting in industrial software development,”

Quality of defect-reports

exactpro EXITUS ACTA PROBAT

Improving defect reports:

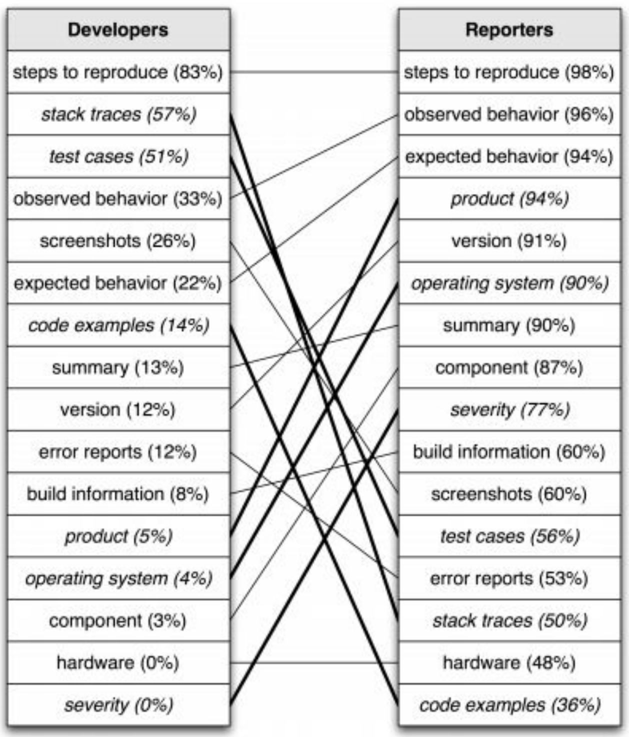
- eliminate user private information from bug-report [1]
- measure comments [2]
- eliminate invalid bug-report [3]
- ways to improve BTS [4]:

1. gathering stack-traces
2. helping users provide better information
3. using automatic defect triage
4. being very clear with the users

[1] M.Castro,M.Costa,andJ.-P.Martin,“Better bug reporting with better privacy”
[2] B. Dit, “Measuring the semantic similarity of comments in bug reports”
[3] J. Sun, “Why are bug reports invalid?”
[4] T. Zimmermann, R. Premraj, J. Sillito, and S. Breu, “Improving bug tracking systems”

Quality of defect-reports

Tools: Cuezilla



Most helpful for developers vs. provided by reporters.

Input data:

1) Action verbs
2) Expected / observed behaviour
3) Steps to reproduce
4) Build-related
5) User interface elements
6) Code samples
7) Stack traces
8) Patches
9) Screenshots
10) Readability

[1] N. Bettenburg, S. Just, A. Schröter, C. Weiss, R. Premraj, and T. Zimmermann, “What makes a good bug report?”

Metrics and prediction of defect reports

Tasks:

- Analysis of defect data
- Predict metrics of testing

Metrics and prediction of defect reports

Analysis of defect data :

- NLP [1]
- Visualize of defect databases [2]
- Automatically generating summaries [3]

[1] K. S. Wasson, K. N. Schmid, R. R. Lutz, and J. C. Knight, “Using occurrence properties of defect report data to improve requirements”
[2] B M. D’Ambros, M. Lanza, and M. Pinzger, ““a bug’s life” visualizing a bug database”
[3] S.Rastkar,G.C.Murphy,andG.Murray,“Summarizing software artifacts:A case study of bug reports”

Metrics and prediction of defect reports

Examples of metrics:

- time to fix / time to resolve[1]
- which defects get reopened [2]
- which defects get fixed [3]
- which defects get rejected

[1] "How long will it take to fix this bug?"; P. Bhattacharya and I. Neamtiu, "Bug-fix time prediction models: Can we do better?"
[2] E.Shihab,A.Ihara,Y.Kamei,W.M.Ibrahim,M.Ohira,B.Adams,A. E. Hassan, and K.-I. Matsumoto, "Predicting re-opened bugs: A case study on the eclipse project"
[3] P. J. Guo, T. Zimmermann, N. Nagappan, and B. Murphy, "Characterizing and predicting which bugs get fixed: An empirical study of microsoft windows"

Metrics and prediction of defect reports

exactpro
EXITUS ACTA PROBAT

Time to resolve -> cheap/expensive bug

Attributes:

- self-reported severity
- readability
- daily load
- submitter reputation
- bug severity changes
- comment count
- attachment count

Open Access Quality Assurance & Related Software Development for Financial Markets
www.exactpro.com
Tel: +7 495 640 24 60 , +1 415 830 38 49

Metrics and prediction of defect reports

Reasons of defect reopening:

- Bug report has insufficient information
- Developers misunderstand the root causes of defect
- Ambiguous requirements in specifications

Using metric allows:

- define weaknesses in testing
- Characterize actual quality of the bug fixing process
- Define weaknesses in documentation

Open Access Quality Assurance & Related Software Development for Financial Markets
www.exactpro.com
Tel: +7 495 640 24 60 , +1 415 830 38 49

Metrics and prediction of defect reports

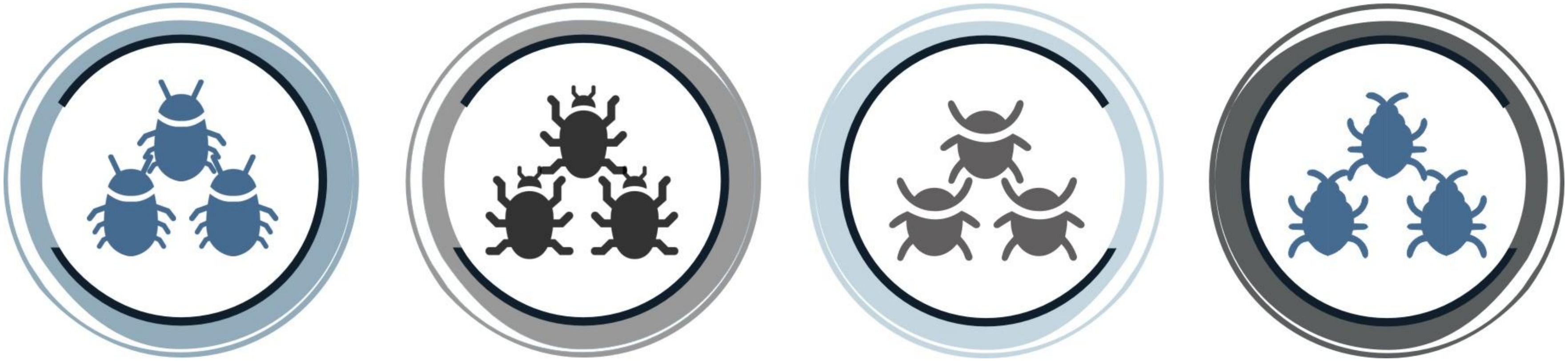
Attributes (reopening of defect):

- Bug source
- Reputation of bug opener
- Reputation of 1st assigner
- Initial severity level
- Severity upgraded?
- Num. editors
- Num. assignee building
- Num. component path changes
- Num. re-opens

Open Access Quality Assurance & Related Software Development for Financial Markets
www.exactpro.com
Tel: +7 495 640 24 60 , +1 415 830 38 49

Defect Management

Defect clustering



- Understand weaknesses of software
- Improve testing strategy

Open Access Quality Assurance & Related Software Development for Financial Markets
www.exactpro.com
Tel: +7 495 640 24 60 , +1 415 830 38 49

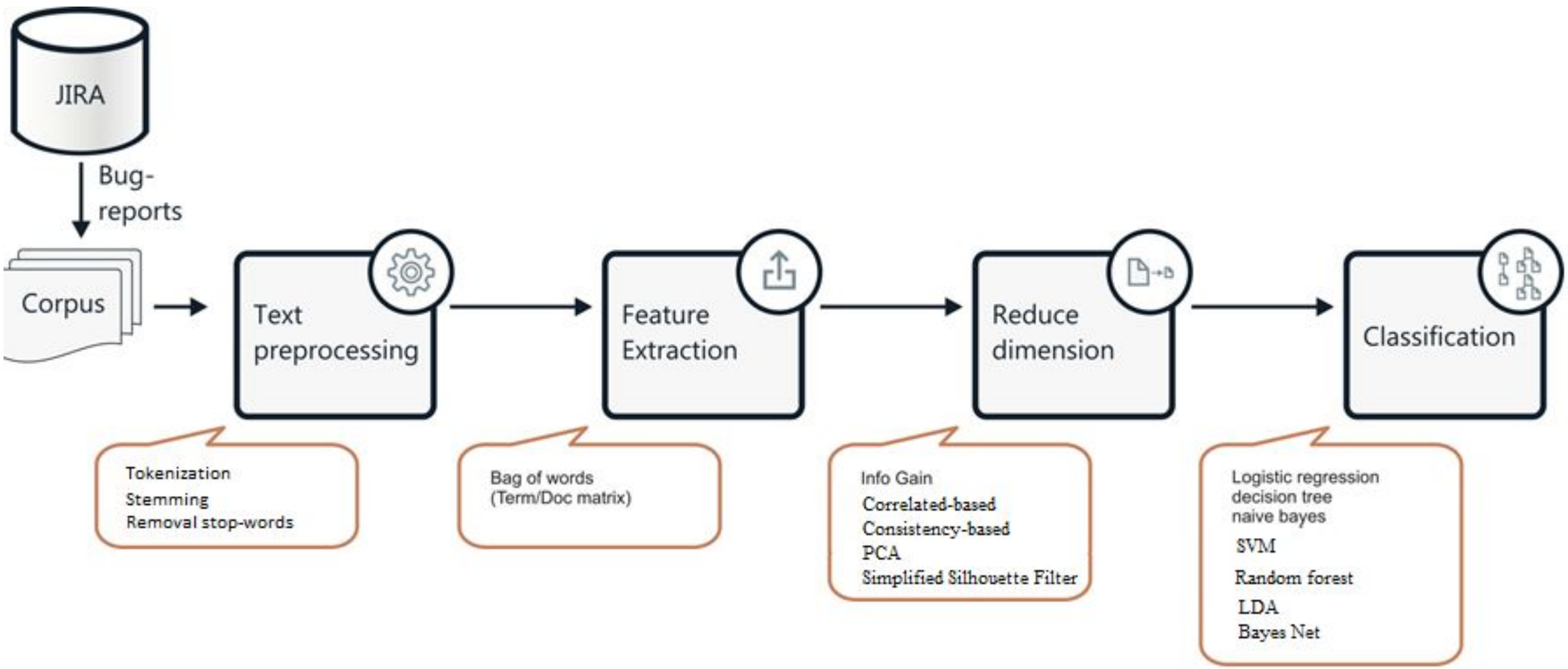
Defect Management

Attributes for cluster analysis:

- Priority
- status
- resolution
- time to resolve
- count of comments
- area of testing

Open Access Quality Assurance & Related Software Development for Financial Markets
www.exactpro.com
Tel: +7 495 640 24 60 , +1 415 830 38 49

Defect Classification



Defect Management

Analyse description utility:

- Stack trace (regular expressions)
- Steps to reproduce (classify)
- Expected/Observed behaviour (classify)
- Readability

Open Access Quality Assurance & Related Software Development for Financial Markets
www.exactpro.com
Tel: +7 495 640 24 60 , +1 415 830 38 49

Defect Management

Attributes for prediction of metric “which defects get reopened”:

- Priority
- status
- resolution
- time to resolve
- count of comments
- count of attach
- description utility

Open Access Quality Assurance & Related Software Development for Financial Markets
www.exactpro.com
Tel: +7 495 640 24 60 , +1 415 830 38 49

Thank you!

Open Access Quality Assurance & Related Software Development for Financial Markets
www.exactpro.com
Tel: +7 495 640 24 60 , +1 415 830 38 49